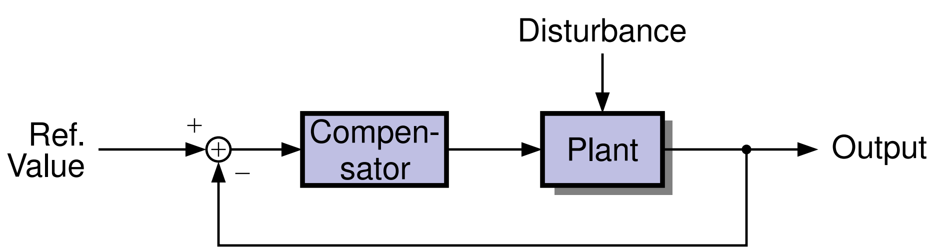
Course Background

1.1: From time to frequency domain

- Loosely speaking, control is the process of getting "something" to do what you want it to do (or "not do," as the case may be).
  - The "something" can be almost anything. Some examples: aircraft, spacecraft, cars, machines, robots, radars, etc.
  - Some less obvious examples: energy systems, the economy, biological systems, the human body. . .
- The "something" is called the system that we would like to control.

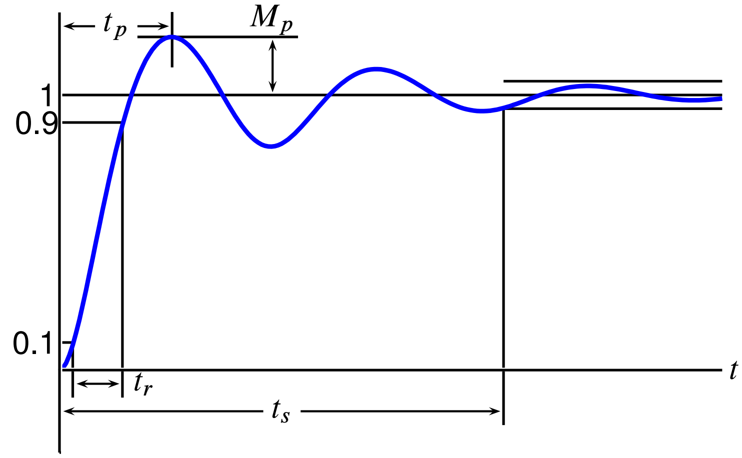
DEFINITION: Control is the process of causing a system variable to conform to some desired value, called a reference value (*e.g.*, variable = temperature for a climate-control system).

DEFINITION: Feedback is the process of measuring the controlled variable and using that information to influence its value.



- Feedback is not necessary for control. But, it is necessary to cater for system uncertainty, which is the principal role of feedback.
- Open-loop control is also possible.

- Goals of feedback control:
  - Change dynamic response of a system to have desired properties.
  - Output of system tracks reference input.
  - Reject disturbances.



- Control design requires mathematical sets of equations (called a <u>model</u>) that describes the system being controlled.
- Classical feedback techniques (*cf.,* ECE4510) use frequency-domain (Laplace) models and tools to analyze and design control systems.
  - Involves moving the <u>poles</u> of the closed-loop <u>transfer function</u>.
- Multivariable, state-space control instead:
  - Primarily uses time-domain matrix representations of systems.
  - Very powerful. Can often place poles of closed-loop system *anywhere we want!* Can make fast, smooth, etc.
  - Same methods work for single-input, single-output (SISO) or multi-input, multi-output (MIMO) systems.
  - Advanced techniques (*cf.,* ECE5530) allow design of *optimal* linear controllers with a single MATLAB command!
- This course is a bridge between classical control and topics in advanced linear systems and control.
- We now review some of the concepts of classical linear systems and control which we will use. . .

Dynamic response

- Our primary objective is to be able to understand and learn how to control linear time-invariant (LTI) systems.
  - We will also spend some time investigating nonlinear and linear time-varying (LTV) systems.
- LTI dynamics may be specified via models expressed as linear, constant-coefficient ordinary differential equations (LCCODE).
- Examples include:
  - Mechanical systems: Use Newton's laws.
  - Electrical systems: Use Kirchoff's laws.
  - Electro-mechanical systems (generator/motor).
  - Thermodynamic systems.
  - Fluid-dynamic systems.

EXAMPLE: Second-order system in "standard form":

$$\ddot{y}(t) + 2\zeta\omega_n\dot{y}(t) + \omega_n^2 y(t) = \omega_n^2 u(t).$$

- $u(t)$ is the input, $y(t)$ is the output, $\dot{y}(t) \triangleq \dfrac{\mathrm{d}y(t)}{\mathrm{d}t}$, and $\ddot{y}(t) \triangleq \dfrac{\mathrm{d}^2 y(t)}{\mathrm{d}t^2}$.

Laplace Transform

- The Laplace transform is a tool to help analyze dynamic systems. $Y(s) = H(s)U(s)$, where
  - $Y(s)$ is Laplace transform of output, $y(t)$;
  - $U(s)$ is Laplace transform of input, $u(t)$;
  - $H(s)$ is transfer function—the Laplace tx of impulse response, $h(t)$.

- $\mathcal{L}\{\dot{y}(t)\} = sY(s) - y(0)$ in general, and $\mathcal{L}\{\dot{y}(t)\} = sY(s)$ for a system initially at rest.

EXAMPLE: Transfer function for second-order system:

$$s^2Y(s) + 2\zeta\omega_n sY(s) + \omega_n^2 Y(s) = \omega_n^2 U(s)$$

$$Y(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} U(s).$$

- Transforms for systems with LCCODE representations can be written as $Y(s) = H(s)U(s)$, where

$$H(s) = \frac{b_0 s^m + b_1 s^{m-1} + \cdots + b_{m-1}s + b_m}{a_0 s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n},$$

where $n \geq m$ for physical systems.

- These can be represented in MATLAB using vectors of numerator and denominator polynomials:

```
num=[b0 b1 ... bm];
den=[a0 a1 ... an];
sys=tf(num,den);
```

- Can also represent these systems by factoring the polynomials into zero-pole-gain form:

$$H(s) = K\frac{\prod_{i=1}^{m}(s - z_i)}{\prod_{i=1}^{n}(s - p_i)}.$$

```
sys=zpk(z,p,k);      % in MATLAB
```

- Input signals of interest include the following:

$$
\begin{aligned}
u(t) &= k\,\delta(t) & \ldots\quad U(s) &= k & &\text{impulse}\\
u(t) &= k\,1(t) & \ldots\quad U(s) &= k/s & &\text{step}\\
u(t) &= kt\,1(t) & \ldots\quad U(s) &= k/s^2 & &\text{ramp}\\
u(t) &= k\exp(-\alpha t)\,1(t) & \ldots\quad U(s) &= \frac{k}{s+\alpha} & &\text{exponential}\\
u(t) &= k\sin(\omega t)\,1(t) & \ldots\quad U(s) &= \frac{k\omega}{s^2+\omega^2} & &\text{sinusoid}\\
u(t) &= k\cos(\omega t)\,1(t) & \ldots\quad U(s) &= \frac{ks}{s^2+\omega^2} & &\text{cosinusoid}\\
u(t) &= ke^{-at}\sin(\omega t)\,1(t) & \ldots\quad U(s) &= \frac{k\omega}{(s+a)^2+\omega^2} & &\text{decaying sinusoid}\\
u(t) &= ke^{-at}\cos(\omega t)\,1(t) & \ldots\quad U(s) &= \frac{k(s+a)}{(s+a)^2+\omega^2} & &\text{decaying cosinusoid}
\end{aligned}
$$

- MATLAB's "`impulse`," "`step`," and "`lsim`" commands can be used to find output time histories.

- The final value theorem states that if a system is stable and has a final, constant value, then

$$\lim_{t\to\infty} x(t) = \lim_{s\to 0} sX(s).$$

- Useful when investigating steady-state errors in a control system.

- The initial value theorem states that the initial value of a signal may be found using

$$\lim_{s\to\infty} sX(s) = \begin{cases} x(0^-) = x(0^+), & x(t) \text{ continuous at } t = 0; \\ x(0^+), & \text{otherwise.} \end{cases}$$

- We will see a use for this later in the semester when studying system controllability.

1.2: From frequency to time domain

- The inverse Laplace transform (ILT) converts $X(s) \to x(t)$.
- Here we assume that $X(s)$ is a ratio of polynomials in s. That is,

$$X(s) = \frac{C(s)}{A(s)}.$$

- If $X(s)$ is not a *proper rational function*, we must first perform long division. (In a proper rational function, the degree of $C(s)$ is less than the degree of $A(s)$.)
- In general, partial-fraction inversion begins by writing

$$X(s) = K(s) + \frac{B(s)}{A(s)} = K(s) + F(s),$$

where $K(s)$ is of the form

$$K(s) = k_0 + k_1 s + \cdots + k_L s^L.$$

- We inverse transform $K(s)$ using

$$\frac{\mathrm{d}^n \delta(t)}{\mathrm{d}t^n} \Longleftrightarrow s^n.$$

- The remaining problem is to find the inverse transform of $F(s)$.

$$\begin{aligned} F(s) &= \frac{b_0 s^m + b_1 s^{m-1} + \cdots + b_m}{s^n + a_a s^{n-1} + \cdots + a_n} \\ &= k \frac{\prod_{i=1}^m (s - z_i)}{\prod_{i=1}^n (s - p_i)} \qquad \Longleftarrow \qquad \frac{\text{(zeros)}}{\text{(poles)}} \\ &= \frac{r_1}{s - p_1} + \frac{r_2}{s - p_2} + \cdots + \frac{r_n}{s - p_n} \qquad \text{if } \{p_i\} \text{ distinct.} \end{aligned}$$

so, $$(s - p_1) F(s) = r_1 + \frac{r_2 (s - p_1)}{s - p_2} + \cdots + \frac{r_n (s - p_1)}{s - p_n}.$$

- Let $s = p_1$. Then,

$$r_1 = (s - p_1)F(s)\big|_{s=p_1}.$$

- Similarly,

$$r_i = (s - p_i)F(s)\big|_{s=p_i}$$

and

$$f(t) = \sum_{i=1}^{n} r_i e^{p_i t} 1(t) \qquad \text{since } \mathcal{L}\left[e^{kt}1(t)\right] = \frac{1}{s-k}.$$

EXAMPLE: $F(s) = \dfrac{5}{s^2 + 3s + 2} = \dfrac{5}{(s+1)(s+2)}$.

$$\begin{aligned} r_1 &= (s+1)F(s)\Big|_{s=-1} = \frac{5}{s+2}\Big|_{s=-1} = 5 \\ r_2 &= (s+2)F(s)\Big|_{s=-2} = \frac{5}{s+1}\Big|_{s=-2} = -5 \\ f(t) &= (5e^{-t} - 5e^{-2t})1(t). \end{aligned}$$

Repeated poles

- If $F(s)$ has repeated roots, we must modify the procedure. *e.g.*, for a pole repeated 3 times:

$$\begin{aligned} F(s) &= \frac{k}{(s-p_1)^3(s-p_2)\cdots} \\ &= \frac{r_{1,1}}{s-p_1} + \frac{r_{1,2}}{(s-p_1)^2} + \frac{r_{1,3}}{(s-p_1)^3} + \frac{r_2}{s-p_2} + \cdots \\ r_{1,3} &= (s-p_1)^3 F(s)\Big|_{s=p_1} \\ r_{1,2} &= \left[\frac{\mathrm{d}}{\mathrm{d}s}\left((s-p_1)^3 F(s)\right)\right|_{s=p_1} \end{aligned}$$

$$r_{1,1} = \frac{1}{2}\left[\frac{\mathrm{d}^2}{\mathrm{d}s^2}\left((s-p_1)^3F(s)\right)\right|_{s=p_1}$$

$$r_{x,k-i} = \frac{1}{i!}\left[\frac{\mathrm{d}^i}{\mathrm{d}s^i}\left((s-p_i)^kF(s)\right)\right|_{s=p_i}.$$

EXAMPLE: Find ILT of $\dfrac{s+3}{(s+1)(s+2)^2}$.

- ans: $f(t) = (2e^{-t} - 2e^{-2t} - \underbrace{te^{-2t}}_{\text{from repeated root.}})1(t).$
- *TEDIOUS.*
- Use MATLAB. *e.g.*, $F(s) = \dfrac{5}{s^2+3s+2}$.

Example 1.

```
>> Fnum=[0 0 5];

>> Fden=[1 3 2];

>> [r,p,k]=residue(Fnum,Fden);

  r = -5

       5

  p = -2

      -1

  k = []
```

Example 2.

```
>> Fnum=[0 0 1 3];

>> Fden=conv([1 1],conv([1 2],[1 2]));

>> [r,p,k]=residue(Fnum,Fden);

  r = -2

      -1

       2

  p = -2

      -2

      -1

  k = []
```

- When you use “residue” and get repeated roots, *BE SURE* to type “help residue” to correctly interpret the result.

Complex-conjugate poles

- The theory developed thus far works for either real or complex poles.

- It may be easier to handle complex-conjugate poles separately.
- Consider

$$F(s) = \frac{B(s)}{(s-p_1)(s-p_1^*)Q(s)}$$
$$= \frac{K_1}{s-p_1} + \frac{K_2}{s-p_1^*} + \frac{R(s)}{Q(s)}.$$

- Expand $R(s)/Q(s)$ using previous methods. Expand the first part as

$$\frac{K_1}{s-p_1} + \frac{K_2}{s-p_1^*} = \frac{as+b}{(s-\sigma_1)^2+\omega_1^2}.$$

- It can be shown that $a = 2\mathbb{R}(K_1)$, $b = -2[\mathbb{R}(K_1)\sigma_1 + \mathbb{I}(K_1)\omega_1]$, $\sigma_1 = \mathbb{R}(p_1)$, and $\omega_1 = \mathbb{I}(p_1)$.

EXAMPLE: As a specific problem consider

$$X(s) = \frac{2s^2+6s+6}{(s+2)(s^2+2s+2)}$$
$$= \frac{r_1}{s+2} + \frac{as+b}{(s+1)^2+1}.$$

- Using the simple-pole formula we find

$$r_1 = \left.\frac{2s^2+6s+6}{s^2+2s+2}\right|_{s=-2} = \frac{8-12+6}{4-4+2} = 1.$$

- We will substitute values for s to obtain a and b.
- Let $s = 0$.

$$\frac{6}{2\cdot 2} = \frac{1}{2} + \frac{b}{2} \;\Rightarrow\; b = 2.$$

- Let $s = 1$.

$$\frac{2+6+6}{3(1+2+2)} = \frac{1}{3} + \frac{a+2}{5} \;\Rightarrow\; a = 1.$$

- Finally,

$$X(s) = \frac{1}{s+2} + \frac{s+2}{(s+1)^2+1}$$
$$= \frac{1}{s+2} + \frac{s+1}{(s+1)^2+1} + \frac{1}{(s+1)^2+1}.$$

- Taking the inverse-Laplace transform

$$x(t) = \left[e^{-2t} + e^{-t}\cos(t) + e^{-t}\sin(t)\right]1(t).$$

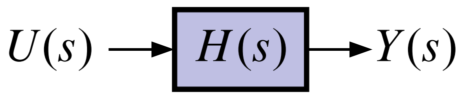
Symbolic Laplace transforms using MATLAB

- MATLAB incorporates part of the Maple symbolic toolbox.
- The commands of interest to us here are: `laplace`, `ilaplace`, `ezplot` and `pretty`.
  - `F=laplace(f)` is the Laplace transform of symbolic fn 'f.'
  - `f=ilaplace(F)` is the inverse-Laplace transform of 'F.'
  - `ezplot(f,[xmin xmax])` plots symbolic function 'f.'
  - `pretty(S)` displays symbolic 'S' in a "pretty" way.

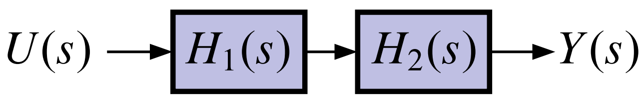
1.3: Dynamic properties of LTI systems

Block diagrams

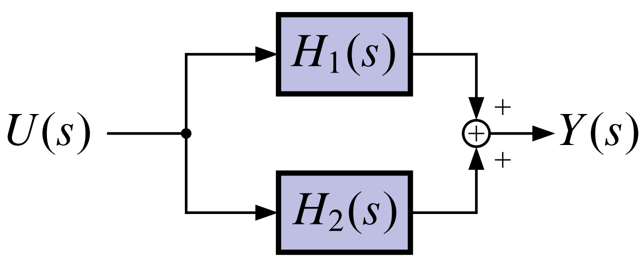
- Useful when analyzing systems comprised of a number of sub-units.



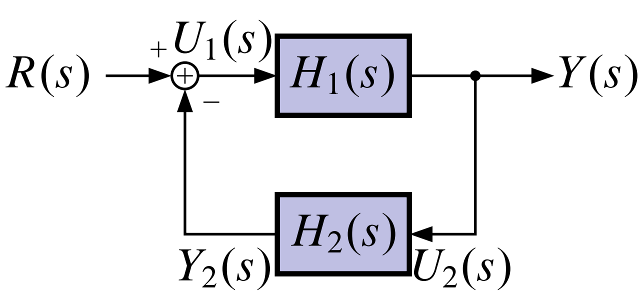
$$Y(s) = H(s)U(s)$$



$$Y(s) = [H_1(s)H_2(s)]\,U(s)$$



$$Y(s) = [H_1(s) + H_2(s)]\,U(s)$$



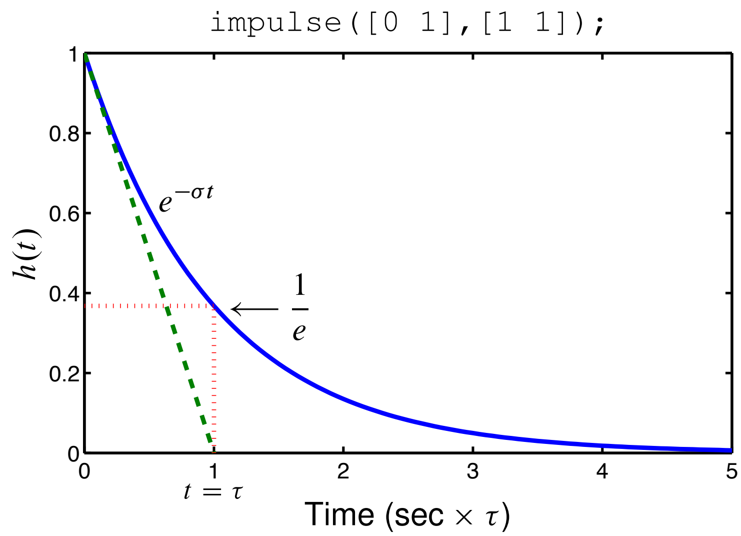
$$Y(s) = \frac{H_1(s)}{1 + H_2(s)H_1(s)}R(s)$$

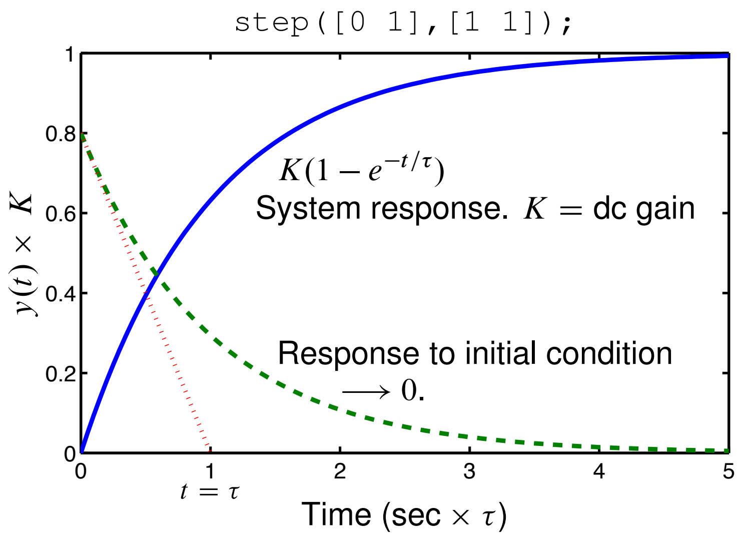
Dynamic response versus pole locations

- The poles of $H(s)$ determine (qualitatively) the dynamic response of the system. The zeros of $H(s)$ quantify the relationship.
- If the system has only real poles, each one is of the form:

$$H(s) = \frac{1}{s + \sigma}.$$

- If $\sigma > 0$, the system is stable, and $h(t) = e^{-\sigma t} 1(t)$. The time constant is $\tau = 1/\sigma$, and the response of the system to an impulse or step decays to steady-state in about 4 or 5 time constants.





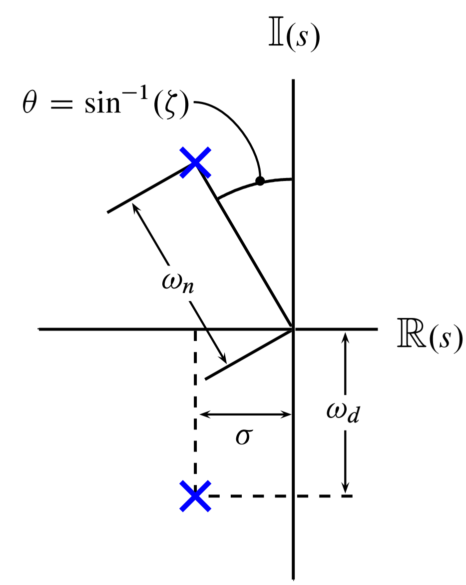
- If a system has complex-conjugate poles, each may be written as:

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}.$$

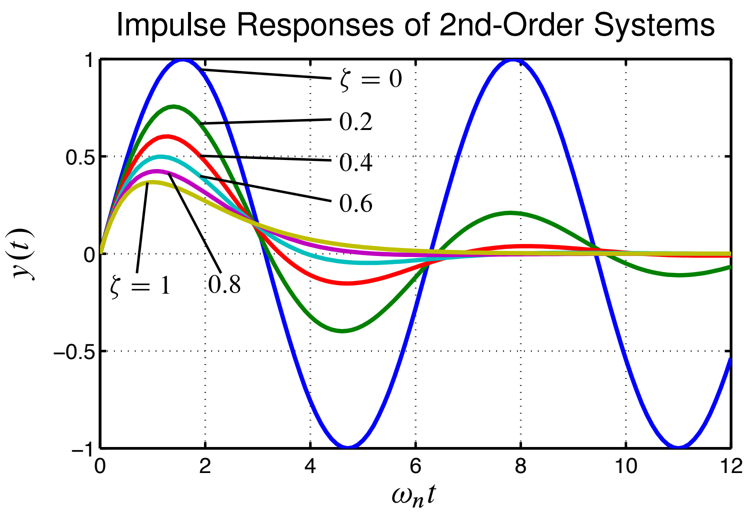
- We can extract two more parameters from this equation:

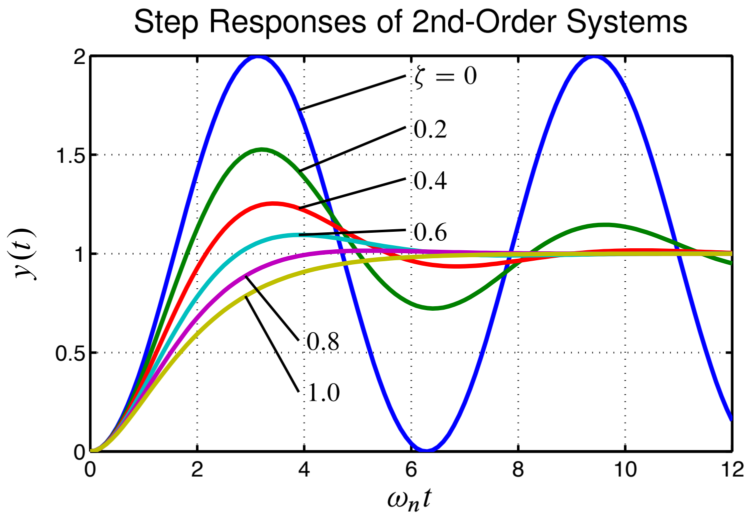
$$\sigma = \zeta\omega_n \quad \text{and} \quad \omega_d = \omega_n\sqrt{1-\zeta^2}.$$

- σ plays the same role as above—it specifies decay rate of the response.
- ω_d is the oscillation frequency of the output. Note: $\omega_d \neq \omega_n$ unless $\zeta = 0$.
- ζ is the "damping ratio" and it also plays a role in decay rate and overshoot.

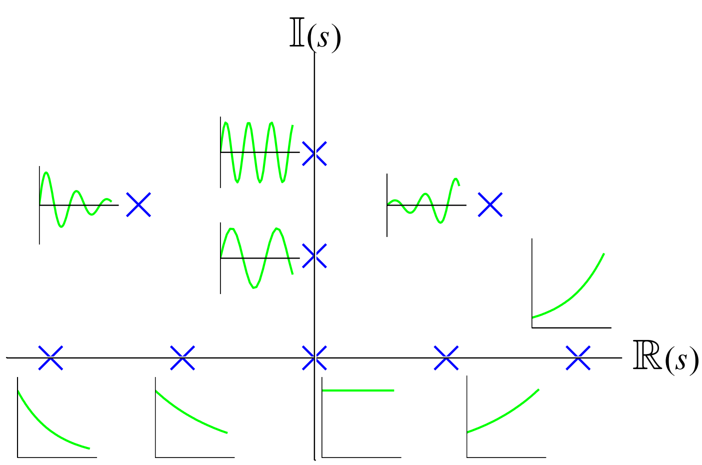


- Impulse response $h(t) = \omega_n e^{-\sigma t} \sin(\omega_d t)\ 1(t)$.
- Step response $y(t) = 1 - e^{-\sigma t}\left(\cos(\omega_d t) + \dfrac{\sigma}{\omega_d}\sin(\omega_d t)\right)$.

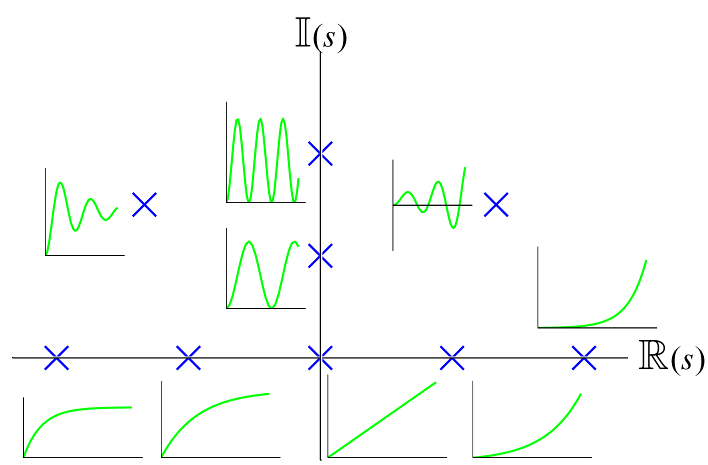




- A summary chart of impulse responses and step responses versus pole locations is:

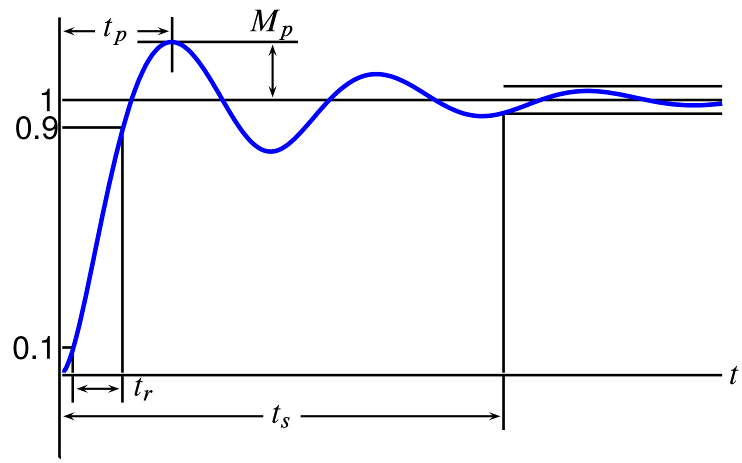


Impulse responses vs. pole locations

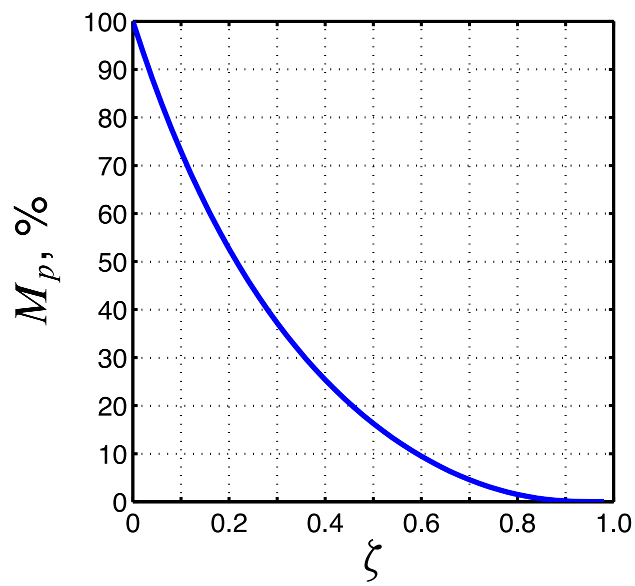


Step responses vs. pole locations

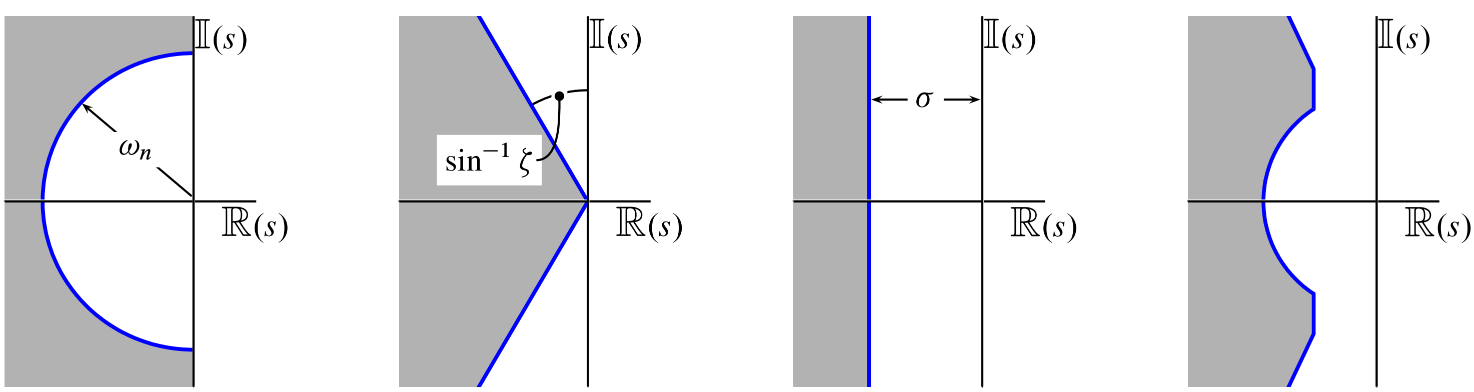
- Time-domain specs. determine where poles *should* be placed in the s-plane.
- Step-response specs often given.



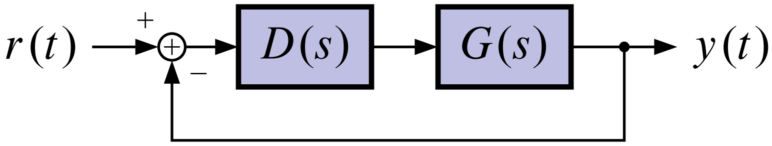
- Rise time t_r = time to go from 10 % to 90 % of final value.
- Settling time t_s = time until permanently within ≈ 1 % of final value.
- Overshoot M_p = maximum *percent* overshoot.



$$
\begin{aligned}
t_r &\approx 1.8/\omega_n & \ldots && \omega_n &\geq 1.8/t_r \\
t_s &\approx 4.6/\sigma & \ldots && \sigma &\geq 4.6/t_s \\
M_p &\approx e^{-\pi\zeta/\sqrt{1-\zeta^2}} & \ldots && \zeta &\geq \text{fn}(M_p)
\end{aligned}
$$



Basic feedback properties



$$\frac{Y(s)}{R(s)} = \frac{D(s)G(s)}{1 + D(s)G(s)} = T(s).$$

- Stability depends on roots of denominator of $T(s)$: $1 + D(s)G(s) = 0$.
- Routh test used to determine stability.
- Steady-state error found from $E(s) = (1 - T(s))\,R(s)$.
- $e_{ss} = \lim_{t\to\infty} e(t) = \lim_{s\to 0} sE(s)$ if the limit exists.
  - System type $= 0$ iff e_{ss} is finite for unit-step reference-input $1(t)$.
  - System type $= 1$ iff e_{ss} is finite for unit-ramp reference-input $r(t)$.
  - System type $= 2$ iff e_{ss} is finite for unit-parabola ref.-input $p(t)\ldots$

Some types of controllers

"Proportional" ctrlr:	$u(t) = Ke(t)$.	$D(s) = K$.
"Integral" ctrlr	$u(t) = \frac{K}{T_I}\int_{-\infty}^{t} e(t)\,\mathrm{d}t$.	$D(s) = \frac{K}{T_I s}$
"Derivative" ctrlr.	$u(t) = KT_D\dot{e}(t)$	$D(s) = KT_D s$
Combinations:	PI: $D(s) = K\left(1 + \frac{1}{T_I s}\right)$;	
	PD: $D(s) = K\,(1 + T_D s)$;	
	PID: $D(s) = K\left(1 + \frac{1}{T_I s} + T_D s\right)$.	
Lead:	$D(s) = K\frac{Ts+1}{\alpha Ts+1}$,	$\alpha < 1$ (approx PD)
Lag:	$D(s) = K\frac{Ts+1}{\alpha Ts+1}$,	$\alpha > 1$ (approx PI; often, $K = \alpha$)

Lead/Lag: $$D(s) = K\frac{(T_1 s + 1)(T_2 s + 1)}{(\alpha_1 T_1 s + 1)(\alpha_2 T_2 s + 1)}, \quad \alpha_1 < 1,\, \alpha_2 > 1.$$

- Integral can reduce/eliminate steady-state error.
- Derivatives can reduce/eliminate oscillation.
- Proportional term can speed/slow response.
- Lead control approximates derivative control, but reduces amplification of noise.
- Lag control approximates integral control, but is easier to stabilize.

Where to from here?

- We have reviewed some important concepts from classical control theory, which uses a transfer-function approach.
- We now begin to examine state-space models.